

10/539846
PCT/JP03/16389

日 本 国 特 許 庁
JAPAN PATENT OFFICE

19.12.03

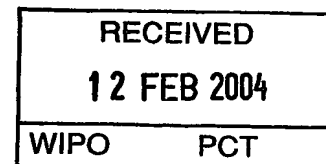
別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 2 年 1 2 月 1 9 日
Date of Application:

出 願 番 号 特 願 2 0 0 2 - 3 6 7 6 0 8
Application Number:
[ST. 10/C] : [J P 2 0 0 2 - 3 6 7 6 0 8]

出 願 人 エヌ・ティ・ティ・コミュニケーションズ株式会社
Applicant(s):

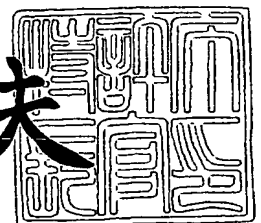


PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

2 0 0 4 年 1 月 2 9 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



BEST AVAILABLE COPY

出証番号 出証特 2 0 0 4 - 3 0 0 3 8 5 3

【書類名】 特許願

【整理番号】 GLN-00388

【提出日】 平成14年12月19日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 17/00

【発明の名称】 データ分割方法および装置

【請求項の数】 2

【発明者】

【住所又は居所】 東京都千代田区内幸町一丁目1番6号 エヌ・ティ・テ
ィ・コミュニケーションズ株式会社内

【氏名】 荻原 利彦

【発明者】

【住所又は居所】 東京都千代田区内幸町一丁目1番6号 エヌ・ティ・テ
ィ・コミュニケーションズ株式会社内

【氏名】 野村 進

【特許出願人】

【識別番号】 399035766

【氏名又は名称】 エヌ・ティ・ティ・コミュニケーションズ株式会社

【代理人】

【識別番号】 100083806

【弁理士】

【氏名又は名称】 三好 秀和

【電話番号】 03-3504-3075

【選任した代理人】

【識別番号】 100068342

【弁理士】

【氏名又は名称】 三好 保男

【選任した代理人】

【識別番号】 100095500

【弁理士】

【氏名又は名称】 伊藤 正和

【選任した代理人】

【識別番号】 100101247

【弁理士】

【氏名又は名称】 高橋 俊一

【選任した代理人】

【識別番号】 100098327

【弁理士】

【氏名又は名称】 高松 俊雄

【手数料の表示】

【予納台帳番号】 001982

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9908855

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 データ分割方法および装置

【特許請求の範囲】

【請求項 1】 元データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するデータ分割方法であって、

元データを処理単位ビット長毎に区分けして、分割数より 1 つ少ない複数の元部分データを生成し、

この複数の元部分データの各々に対応して処理単位ビット長の複数の乱数部分データを生成し、

各分割データを処理単位ビット長毎に区分けして分割数より 1 つ少ない複数の分割部分データを生成した場合の各分割部分データを元部分データと乱数部分データの排他的論理和を含む所定の定義式に従って生成すること

を特徴とするデータ分割方法。

【請求項 2】 元データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するデータ分割装置であって、

元データを処理単位ビット長毎に区分けして、分割数より 1 つ少ない複数の元部分データを生成する元部分データ生成手段と、

この複数の元部分データの各々に対応して処理単位ビット長の複数の乱数部分データを生成する乱数生成手段と、

各分割データを処理単位ビット長毎に区分けして分割数より 1 つ少ない複数の分割部分データを生成した場合の各分割部分データを元部分データと乱数部分データの排他的論理和を含む所定の定義式に従って生成する分割部分データ生成手段と

を有することを特徴とするデータ分割装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、データの機密性および安全性を確保するためにはデータを分割して保管することが有効であるが、このような場合などに有効なデータ分割方法およ

び装置に関し、更に詳しくは、元データを所望の処理単位ビットに基づいて所望の分割数の分割データに分割するデータ分割方法および装置に関する。

【0002】

【従来の技術】

重要な秘密データ（以下、元データという）を保管する場合、紛失、破壊、盗難やプライバシー侵害の脅威がある。このような脅威は秘密に保管すべきデータを単に暗号化しただけでは解決できず、紛失、破壊に備えてコピーを複数作ることが有効であるが、コピーを複数作ると盗難のリスクが増加してしまう。

【0003】

このような問題を解決する手段として、従来、しきい値秘密分散法がある。この従来の方法は、元データSをn個のデータに分割し、そのうち任意のx個の分割データを集めれば元データSが復元できるが、任意のx-1個の分割データでは元データSは復元できないというものである。従って、x-1個まで分割データが盗まれても元データSが漏れず、またn-x個まで分割データを紛失したり破壊されたりしても、元データSを復元できる。

【0004】

この方法の代表的な実現例としてm-1次多項式と剰余演算により構成される方法がある（非特許文献2参照）。この従来の方法は、公開鍵暗号方式の秘密鍵の分割管理などで利用されており、データ量が1キロバイトに満たない程度であるため、現状のコンピュータの演算処理能力、記憶装置・記憶媒体などのコストに対しては特に問題ない。

【0005】

【非特許文献1】

Shamir, A "How to Share a Secret" Comm. Assoc. Comput. Mach., vol. 22, no. 11, pp. 612-613 (Nov. 1979)

【非特許文献2】

Bruce Schneier "Applied Cryptography", John Wiley & Sons, Inc., pp. 383-384 (1994)

【0006】

【発明が解決しようとする課題】

上述した従来の方法を安全に保管したいデータ量が例えばメガバイト、ギガバイトまたはそれ以上の規模となった場合に利用すると、多項式演算・剰余演算などを含む多倍長整数の演算処理を大量のデータに対して行う演算処理能力が必要となるとともに、またこの従来の方法では、例えば分割数 $n=5$ の場合には1バイトのデータから1バイトの分割データが5つ生成されるため、元データに対して単純に分割数に比例した倍数の記憶容量が必要となるなど、コンピュータを用いて具体的に実現する上で現実的ではないという問題がある。

【0007】

また、上述した従来の方法では、データの機密性を確保するため分割演算のための処理単位を設定しているが、この分割演算の処理単位にある程度のデータ長が必要となり、任意の処理単位で分割演算を行うことができないという問題もある。

【0008】

本発明は、上記に鑑みてなされたもので、その目的とするところは、比較的簡単な処理により元データを効率的に分割し得るデータ分割方法および装置を提供することにある。

【0009】**【課題を解決するための手段】**

上記目的を達成するため、請求項1記載の本発明は、元データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するデータ分割方法であって、元データを処理単位ビット長毎に区分けして、分割数より1つ少ない複数の元部分データを生成し、この複数の元部分データの各々に対応して処理単位ビット長の複数の乱数部分データを生成し、各分割データを処理単位ビット長毎に区分けして分割数より1つ少ない複数の分割部分データを生成した場合の各分割部分データを元部分データと乱数部分データの排他的論理和を含む所定の定義式に従って生成することを要旨とする。

【0010】

請求項1記載の本発明にあつては、元データを処理単位ビット長毎に区分けし

て複数の元部分データを生成し、複数の乱数部分データを生成し、各分割データを構成する各分割部分データを元部分データと乱数部分データの排他的論理和からなる所定の定義式に従って生成するため、従来のように多項式や剰余演算を用いることなく、コンピュータ処理に適したビット演算である排他的論理和演算を用いることにより高速かつ高性能な演算処理能力を必要とせず、大容量のデータに対しても簡単な演算処理を繰り返して分割データを簡単かつ迅速に生成することができるとともに、また分割データの保管に必要となる記憶容量も分割数に比例した倍数の容量よりも小さくすることができる。

【0011】

また、前記所定の定義式が、元データ、乱数、分割データ、分割数および処理単位ビット長をそれぞれS, R, D, nおよびbで表すとともに、複数n個のうちの1つを表わす変数としてi(=1~n)およびj(=1~n-1)を用いて複数(n-1)個の元部分データ、複数(n-1)個の乱数部分データ、複数(n)個の分割データおよび各分割データの複数(n-1)個の分割部分データのそれぞれのうちの1つをそれぞれS(i), R(i), D(i)およびD(i, j)で表わし、変数iを1からn-1まで変えて、各元部分データS(i)を元データSのb×(i-1)+1ビット目からbビット分のデータとして作成し、U[n, n]をn×n行列である上三角行列とし、P[n, n]をn×n行列である回転行列としたとき、c(j, i, k)を(n-1)×(n-1)行列であるU[n-1, n-1]×(P[n-1, n-1])^(j-1)のi行k列の値と定義し、c(j, i, k)=1のとき、Q(j, i, k)=R(k), c(j, i, k)=0のとき、Q(j, i, k)=0と定義したとき、変数iを1からnまで変えながら、各変数iにおいて変数jを1からn-1まで変えた場合において、i<nのとき、各分割部分データD(i, j)を

【数1】

$$D(i, j) = S(j) * \left(\prod_{k=1}^{n-1} Q(j, i, k) \right)$$

と設定し、i=nのとき、各分割部分データD(i, j)を

$$D(i, j) = R(j)$$

と設定し、上記操作を元データSの先頭から末尾まで繰り返し行うことにより分割数nの分割データを生成することを要旨としても良い。

【0012】

この場合にあつては、所定の定義式が元部分データと乱数部分データの排他的論理和からなるため、従来のように多項式や剰余演算を行う高速かつ高性能な演算処理能力を必要とせず、大容量のデータに対しても簡単な演算処理を繰り返して分割データを簡単かつ迅速に生成することができる。

【0013】

また、請求項2記載の本発明は、元データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するデータ分割装置であつて、元データを処理単位ビット長毎に区分けして、分割数より1つ少ない複数の元部分データを生成する元部分データ生成手段と、この複数の元部分データの各々に対応して処理単位ビット長の複数の乱数部分データを生成する乱数生成手段と、各分割データを処理単位ビット長毎に区分けして分割数より1つ少ない複数の分割部分データを生成した場合の各分割部分データを元部分データと乱数部分データの排他的論理和を含む所定の定義式に従って生成する分割部分データ生成手段とを有することを要旨とする。

【0014】

請求項2記載の本発明にあつては、元データを処理単位ビット長毎に区分けして複数の元部分データを生成し、複数の乱数部分データを生成し、各分割データを構成する各分割部分データを元部分データと乱数部分データの排他的論理和からなる所定の定義式に従って生成するため、従来のように多項式や剰余演算を用いることなく、コンピュータ処理に適したビット演算である排他的論理和演算を用いることにより高速かつ高性能な演算処理能力を必要とせず、大容量のデータに対しても簡単な演算処理を繰り返して分割データを簡単かつ迅速に生成することができるとともに、また分割データの保管に必要となる記憶容量も分割数に比例した倍数の容量よりも小さくすることができる。

【0015】

また、前記所定の定義式が、元データ、乱数、分割データ、分割数および処理単位ビット長をそれぞれ S, R, D, n および b で表すとともに、複数 n 個のうちの1つを表わす変数として i ($=1 \sim n$) および j ($=1 \sim n-1$) を用いて複数 $(n-1)$ 個の元部分データ、複数 $(n-1)$ 個の乱数部分データ、複数 (n) 個の分割データおよび各分割デー

タの複数(n-1)個の分割部分データのそれぞれのうちの1つをそれぞれS(j), R(j), D(i)およびD(i, j)で表わし、変数iを1からn-1まで変えて、各元部分データS(j)を元データSのb×(j-1)+1ビット目からbビット分のデータとして作成し、U[n, n]をn×n行列である上三角行列とし、P[n, n]をn×n行列である回転行列としたとき、c(j, i, k)を(n-1)×(n-1)行列である $U[n-1, n-1] \times (P[n-1, n-1])^{(j-1)}$ のi行k列の値と定義し、c(j, i, k)=1のとき、Q(j, i, k)=R(k), c(j, i, k)=0のとき、Q(j, i, k)=0と定義したとき、変数iを1からnまで変えながら、各変数iにおいて変数jを1からn-1まで変えた場合において、i<nのとき、各分割部分データD(i, j)を

【数2】

$$D(i, j) = S(j) * \left(\prod_{k=1}^{n-1} Q(j, i, k) \right)$$

と設定し、i=nのとき、各分割部分データD(i, j)を

$$D(i, j) = R(j)$$

と設定し、上記操作を元データSの先頭から末尾まで繰り返し行うことにより分割数nの分割データを生成することを要旨とする。

【0016】

この場合にあっては、所定の定義式が元部分データと乱数部分データの排他的論理和からなるため、従来のように多項式や剰余演算を行う高速かつ高性能な演算処理能力を必要とせず、大容量のデータに対しても簡単な演算処理を繰り返して分割データを簡単かつ迅速に生成することができる。

【0017】

【発明の実施の形態】

以下、図面を用いて本発明の実施の形態を説明する。図1は、本発明の一実施形態に係るデータ分割方法を実施するデータ分割装置を含むシステム構成図である。本実施形態のデータ分割装置は、符号1で示すように分割装置1としてネットワーク3に接続されて設けられ、このネットワーク3にアクセスしてくる端末5からの元データ分割要求に応じて元データSを複数の分割データに分割し、この分割した複数の分割データをネットワーク3を介して複数の保管サーバ7a, 7b, 7cに保管するようになっている。なお、図1では、分割装置1は、端末

5 からの元データSを3つの分割データD(1),D(2),D(3)に分割し、それぞれを複数の保管サーバ7 a, 7 b, 7 cに保管するようにしている。

【0018】

また、分割装置1は、ネットワーク3を介してアクセスしてくる端末5からの元データ復元要求に応じて複数の分割データD(1),D(2),D(3)をネットワーク3を介して各保管サーバ7から取得し、この取得した複数の分割データD(1),D(2),D(3)から元データSを復元し、ネットワーク3を介して端末5に送信するようになっている。

【0019】

分割装置1は、詳しくは、元データSから複数の分割データDを生成する分割データ生成手段11、複数の分割データDから元データSを復元する元データ復元手段13、元データSから複数の分割データDを生成するために使用される乱数Rを発生する乱数発生手段15、および分割データ生成手段11で生成した複数の分割データDをネットワーク3を介して複数の保管サーバ7 a, 7 b, 7 cに送信したり、また複数の保管サーバ7 a, 7 b, 7 cからの複数の分割データDをネットワーク3を介して受信するためのデータ送受信手段17から構成されている。

【0020】

上述したように構成される本実施形態における元データの分割および復元では、元データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するが、この場合の処理単位ビット長は任意の値に設定することができ、元データを処理単位ビット長毎に区分けして、この元部分データから分割部分データを分割数より1少ない数ずつ生成するので、元データのビット長が処理単位ビット長の(分割数-1)倍の整数倍に一致しない場合は、元データの末尾の部分に0を埋めるなどして元データのビット長を処理単位ビット長の(分割数-1)倍の整数倍に合わせることで本実施形態を適用することができる。

【0021】

また、上述した乱数も(分割数-1)個の元部分データの各々に対応して処理単位ビット長のビット長を有する(分割数-1)個の乱数部分データとして乱数発生

手段 15 から生成される。すなわち、乱数は処理単位ビット長毎に区分けされて、処理単位ビット長のビット長を有する（分割数-1）個の乱数部分データとして生成される。更に、元データは処理単位ビット長に基づいて所望の分割数の分割データに分割されるが、この分割データの各々も（分割数-1）個の元部分データの各々に対応して処理単位ビット長のビット長を有する（分割数-1）個の分割部分データとして生成される。すなわち、分割データの各々は、処理単位ビット長毎に区分けされて、処理単位ビット長のビット長を有する（分割数-1）個の分割部分データとして生成される。

【0022】

なお、以下の説明では、上述した元データ、乱数、分割データ、分割数および処理単位ビット長をそれぞれ S, R, D, n および b で表すとともに、また複数のデータや乱数などのうちの 1 つを表わす変数として $i (=1 \sim n)$ および $j (=1 \sim n-1)$ を用い、（分割数 $n-1$ ）個の元部分データ、（分割数 $n-1$ ）個の乱数部分データ、および分割数 n 個の分割データ D のそれぞれのうちの 1 つをそれぞれ $S(j), R(j)$ および $D(i)$ で表記し、更に各分割データ $D(i)$ を構成する複数 $(n-1)$ の分割部分データを $D(i, j)$ で表記するものとする。すなわち、 $S(j)$ は、元データ S の先頭から処理単位ビット長毎に区分けして 1 番から順に採番した時の j 番目の元部分データを表すものである。

【0023】

この表記を用いると、元データ、乱数データ、分割データとこれらをそれぞれ構成する元部分データ、乱数部分データ、分割部分データは、次のように表記される。

【0024】

【数 3】

元データ $S = (n-1)$ 個の元部分データ $S(j)$
 $= S(1), S(2), \dots, S(n-1)$

乱数 $R = (n-1)$ 個の乱数部分データ $R(j)$
 $= R(1), R(2), \dots, R(n-1)$

n 個の分割データ $D(i) = D(1), D(2), \dots, D(n)$

各分割部分データ $D(i, j)$

$=D(1, 1), D(1, 2), \dots, D(1, n-1)$

$D(2, 1), D(2, 2), \dots, D(2, n-1)$

$\dots \quad \dots \quad \dots$

$D(n, 1), D(n, 2), \dots, D(n, n-1)$

$(i=1 \sim n), (j=1 \sim n-1)$

本実施形態は、上述したように処理単位ビット長毎に区分けされる複数の部分データに対して元部分データと乱数部分データの排他的論理和演算 (XOR) を行って、詳しくは、元部分データと乱数部分データの排他的論理和演算 (XOR) からなる定義式を用いて、元データの分割を行うことを特徴とするものであり、上述したデータ分割処理に多項式や剰余演算を用いる従来の方法に比較して、コンピュータ処理に適したビット演算である排他的論理和 (XOR) 演算を用いることにより高速かつ高性能な演算処理能力を必要とせず、大容量のデータに対しても簡単な演算処理を繰り返して分割データを生成することができるとともに、また分割データの保管に必要となる記憶容量も分割数に比例した倍数の容量よりも小さくすることができる。更に、任意に定めた一定の長さ毎にデータの先頭から順に演算処理を行うストリーム処理により分割データが生成される。

【0025】

なお、本実施形態で使用する排他的論理和演算 (XOR) は、以下の説明では、「*」なる演算記号で表すことにするが、この排他的論理和演算のビット毎の演算規則での各演算結果は下記のとおりである。

【0026】

$0 * 0$ の演算結果は 0

$0 * 1$ の演算結果は 1

$1 * 0$ の演算結果は 1

$1 * 1$ の演算結果は 0

また、XOR演算は交換法則、結合法則が成り立つ。すなわち、

$$a * b = b * a$$

$$(a * b) * c = a * (b * c)$$

が成り立つことが数学的に証明される。

【0027】

また、 $a*a=0$, $a*0=0*a=a$ が成り立つ。

【0028】

ここで a, b, c は同じ長さのビット列を表し、 0 はこれらと同じ長さですべて「0」からなるビット列を表す。

【0029】

次に、フローチャートなどの図面も参照して、上記実施形態の作用について説明するが、この説明の前に図2、図5、図8、図9のフローチャートに示す記号の定義について説明する。

【0030】

(1) $\prod_{i=1}^n A(i)$ は、 $A(1)*A(2)*\dots*A(n)$ を意味するものとする。

【0031】

(2) $c(j, i, k)$ を $(n-1) \times (n-1)$ 行列である。 $U[n-1, n-1] \times (P[n-1, n-1])^{(j-1)}$ の i 行 k 列の値と定義する。

【0032】

このとき $Q(j, i, k)$ を下記のように定義する。

【0033】

$c(j, i, k)=1$ のとき $Q(j, i, k)=R((n-1) \times m+k)$

$c(j, i, k)=0$ のとき $Q(j, i, k)=0$

(3) $U[n, n]$ とは、 $n \times n$ 行列であって、 i 行 j 列の値を $u(i, j)$ で表すと、

$i+j \leq n+1$ のとき $u(i, j)=1$

$i+j > n+1$ のとき $u(i, j)=0$

である行列を意味するものとし、「上三角行列」ということとする。具体的には下記のような行列である。

【0034】

【数 4】

$$U[3, 3] = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad U[4, 4] = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

(4) $P[n, n]$ とは、 $n \times n$ 行列であって、 i 行 j 列の値を $p(i, j)$ で表すと、

$j=i+1$ のとき $p(i, j)=1$

$i=1, j=n$ のとき $p(i, j)=1$

上記以外のとき $p(i, j)=0$

である行列を意味するものとし、「回転行列」ということとする。具体的には下記のような行列であり、他の行列の右側からかけると当該他の行列の1列目を2列目へ、2列目を3列目へ、 \dots , $n-1$ 列目を n 列目へ、 n 列目を1列目へ移動させる作用がある。つまり、行列 P を他の行列に右側から複数回かけると、その回数分だけ各列を右方向へ回転させるように移動させることができる。

【0035】

【数 5】

$$P[3, 3] = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \quad U[4, 4] = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

(5) A, B を $n \times n$ 行列とすると、 $A \times B$ とは行列 A と B の積を意味するものとする。行列の成分同士の計算規則は通常の数学で用いるものと同じである。

【0036】

(6) A を $n \times n$ 行列とし、 i を整数とすると、 A^i とは行列 A の i 個の積を意味するものとする。また、 A^0 とは単位行列 E を意味するものとする。

【0037】

(7) 単位行列 $E[n, n]$ とは、 $n \times n$ 行列であって、 i 行 j 列の値を $e(i, j)$ で表すと

$i=j$ のとき $e(i, j)=1$

上記以外るとき $e(i, j)=0$

である行列を意味するものとする。具体的には下記のような行列である。Aを任意の $n \times n$ 行列とすると

$$A \times E = E \times A = A$$

となる性質がある。

【0038】

【数6】

$$E[3, 3] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad E[4, 4] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

次に、図2に示すフローチャートおよび図3、図4に示す具体的データなどを参照して、上記実施形態の作用として、まず元データSの分割処理について説明する。

【0039】

本実施形態の分割装置1の利用者は、端末5からネットワーク3を介して分割装置1にアクセスし、分割装置1に元データSを送信し、分割装置1ではデータ送受信手段17が端末5からの元データSを受信し、分割装置1に供給する（図2のステップS201）。なお、本例では、元データSは、16ビットの「10110010 00110111」とする。

【0040】

次に、利用者は端末5から分割数 n として3を分割装置1に指示する（ステップS203）。この分割数 n は分割装置1において予め定められた値を用いてもよい。なお、この分割数 $n=3$ に従って分割装置1で生成される3個の分割データを $D(1)$ 、 $D(2)$ 、 $D(3)$ とする。この分割データ $D(1)$ 、 $D(2)$ 、 $D(3)$ は、すべて元データのビット長と同じ16ビット長のデータである。

【0041】

それから、元データSを分割するために使用される処理単位ビット長bを8ビットと決定し、また元データと同じビット長である16ビットの乱数Rを乱数発生手段15から取得して生成する（ステップS205）。この処理単位ビット長bは、利用者が端末5から分割装置1に対して指定してもよいし、または分割装置1において予め定められた値を用いてもよい。なお、処理単位ビット長bは、任意のビット数でよいが、ここでは元データSを割り切れることができる8ビットとしている。従って、上記16ビットの「10110010 00110111」の元データSは、8ビットの処理単位ビット長で分けられた場合の2個の元分割データS(1)およびS(2)は、それぞれ「10110010」および「00110111」となる。

【0042】

次のステップS207では、元データSのビット長が 8×2 の整数倍であるか否かを判定し、整数倍でない場合には、元データSの末尾を0で埋めて、 8×2 の整数倍に合わせる。なお、本例のように処理単位ビット長bが8ビットおよび分割数nが3に設定された場合における分割処理は、元データSのビット長として16ビットに限られるものでなく、処理単位ビット長 $b \times (\text{分割数}n - 1) = 8 \times 2$ の整数倍の元データSに対して有効なものである。

【0043】

次に、ステップS209では、変数m、すなわち上述した整数倍を意味する変数mを0に設定する。本例のように、元データSが処理単位ビット長 $b \times (\text{分割数}n - 1) = 8 \times 2 = 16$ ビットである場合には、変数mは0であるが、2倍の32ビットの場合には、変数mは1となり、3倍の48ビットの場合には、変数mは2となる。

【0044】

次に、元データSの $8 \times 2 \times m + 1$ ビット目から 8×2 ビット分のデータが存在するか否かが判定される（ステップS211）。これは、このステップS211以降に示す分割処理を元データSの変数mで特定される処理単位ビット長 $b \times (\text{分割数}n - 1) = 8 \times 2 = 16$ ビットに対して行った後、元データSとして次の16ビットがあるか否かを判定しているものである。本例のように元データSが16ビットである場合には、16ビットの元データSに対してステップS211以降の分割処理を1回

行くと、後述するステップS 2 1 9で変数 m が+1されるが、本例の元データSでは変数 m が $m+1$ の場合に相当する17ビット以降のデータは存在しないので、ステップS 2 1 1からステップS 2 2 1に進むことになるが、今の場合は、変数 m は0であるので、元データSの $8 \times 2 \times m + 1$ ビット目は、 $8 \times 2 \times 0 + 1 = 1$ となり、元データSの16ビットの1ビット目から 8×2 ビット分にデータが存在するため、ステップS 2 1 3に進む。

【0045】

ステップS 2 1 3では、変数 j を1から2(=分割数 $n-1$)まで変えて、元データSの $8 \times (2 \times m + j - 1) + 1$ ビット目から8ビット分(=処理単位ビット長)のデータを元部分データS($2 \times m + j$)に設定し、これにより元データSを処理単位ビット長で分けした2(分割数 $n-1$)個の元部分データS(1), S(2)を次のように生成する。

【0046】

元データS=S(1), S(2)

第1の元部分データS(1)=[10110010]

第2の元部分データS(2)=[00110111]

次に、変数 j を1から2(=分割数 $n-1$)まで変えて、乱数部分データR($2 \times m + j$)に乱数発生手段15から発生する8ビットの長さの乱数を設定し、これにより乱数Rを処理単位ビット長で分けした2(分割数 $n-1$)個の乱数部分データR(1), R(2)を次のように生成する(ステップS 2 1 5)。

【0047】

乱数R=R(1), R(2)

第1の乱数部分データR(1)=[10110001]

第2の乱数部分データR(2)=[00110101]

次に、ステップS 2 1 7において、変数 i を1から3(=分割数 n)まで変えるとともに、更に各変数 i において変数 j を1から2(=分割数 $n-1$)まで変えながら、ステップS 2 1 7に示す分割データを生成するための元部分データと乱数部分データの排他的論理和からなる定義式により複数の分割データD(i)の各々を構成する各分割部分データD($i, 2 \times m + j$)を生成する。この結果、次に示すような分割データDが生成される。

【0048】

【数7】

分割データD

 $= 3 \text{ 個の分割データ } D(i)=D(1), D(2), D(3)$

第1の分割データD(1)

 $= 2 \text{ 個の分割部分データ } D(1, j)=D(1, 1), D(1, 2)$ $= \text{「00110110」}, \text{「10110011」}$

第2の分割データD(2)

 $= 2 \text{ 個の分割部分データ } D(2, j)=D(2, 1), D(2, 2)$ $= \text{「00000011」}, \text{「00000010」}$

第3の分割データD(3)

 $= 2 \text{ 個の分割部分データ } D(3, j)=D(3, 1), D(3, 2)$ $= \text{「10110001」}, \text{「00110101」}$

なお、各分割部分データ(i, j)を生成するためのステップS 2 1 7に示す定義式は、本例のように分割数n=3の場合には、具体的には図4に示す表に記載されているものとなる。図4に示す表から、分割部分データD(1, 1)を生成するための定義式は $S(1)*R(1)*R(2)$ であり、D(1, 2)の定義式は $S(2)*R(1)*R(2)$ であり、D(2, 1)の定義式は $S(1)*R(1)$ であり、D(2, 2)の定義式は $S(2)*R(2)$ であり、D(3, 1)の定義式は $R(1)$ であり、D(3, 2)の定義式は $R(2)$ である。また、図4に示す表には $m>0$ の場合の任意の整数についての一般的な定義式も記載されている。

【0049】

このように整数倍を意味する変数 $m=0$ の場合について分割データDを生成した後、次に変数 m を1増やし(ステップS 2 1 9)、ステップS 2 1 1に戻り、変数 $m+1$ に該当する元データSの17ビット以降について同様の分割処理を行おうとするが、本例の元データSは16ビットであり、17ビット以降のデータは存在しないので、ステップS 2 1 1からステップS 2 2 1に進み、上述したように生成した分割データD(1), D(2), D(3)を分割装置1のデータ送受信手段17からネットワーク3を介して保管サーバ7a, 7b, 7cにそれぞれ送信し、各保管サーバ7に保管し、分割処理を終了する。なお、このように保管された分割データD(1)

,D(2),D(3)はそれぞれ単独では元データが推測できない。

【0050】

ここで、上述した図2のフローチャートのステップS217における定義式による分割データの生成処理、具体的には分割数 $n=3$ の場合の分割データの生成処理について詳しく説明する。

【0051】

まず、整数倍を意味する変数 $m=0$ の場合には、ステップS217に示す定義式から各分割データ $D(i)=D(1) \sim D(3)$ の各々を構成する各分割部分データ $D(i, 2 \times m + j)=D(i, j)$ ($i=1 \sim 3, j=1 \sim 2$)は、次のようになる。

【0052】

$$D(1, 1)=S(1)*Q(1, 1, 1)*Q(1, 1, 2)$$

$$D(1, 2)=S(2)*Q(2, 1, 1)*Q(2, 1, 2)$$

$$D(2, 1)=S(1)*Q(1, 2, 1)*Q(1, 2, 2)$$

$$D(2, 2)=S(2)*Q(2, 2, 1)*Q(2, 2, 2)$$

$$D(3, 1)=R(1)$$

$$D(3, 2)=R(2)$$

上記の6つの式のうち上から4つの式に含まれる $Q(j, i, k)$ を具体的に求める。これは $c(j, i, k)$ を 2×2 行列である $U[2, 2] \times (P[2, 2])^{(j-1)}$ の i 行 k 列の値としたとき下記のように定義される。

【0053】

$$c(j, i, k)=1 \text{ のとき } Q(j, i, k)=R(k)$$

$$c(j, i, k)=0 \text{ のとき } Q(j, i, k)=0$$

ここで、

$j=1$ のときは

【数 8】

$$\begin{aligned}
 U[2, 2] \times (P[2, 2])^{-(j-1)} &= U[2, 2] \times (P[2, 2])^{-0} \\
 &= U[2, 2] \times E[2, 2] \\
 &= U[2, 2] \\
 &= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}
 \end{aligned}$$

j=2のときは

【数 9】

$$\begin{aligned}
 U[2, 2] \times (P[2, 2])^{-(j-1)} &= U[2, 2] \times (P[2, 2])^{-1} \\
 &= U[2, 2] \times P[2, 2] \\
 &= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\
 &= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}
 \end{aligned}$$

これを用いると、各分割部分データD(i, j)は次のような定義式により生成される。

【0054】

【数10】

$$\begin{aligned}
 D(1, 1) &= S(1) * Q(1, 1, 1) * Q(1, 1, 2) = S(1) * R(1) * R(2) \\
 D(1, 2) &= S(2) * Q(2, 1, 1) * Q(2, 1, 2) = S(2) * R(1) * R(2) \\
 D(2, 1) &= S(1) * Q(1, 2, 1) * Q(1, 2, 2) = S(1) * R(1) * 0 = S(1) * R(1) \\
 D(2, 2) &= S(2) * Q(2, 2, 1) * Q(2, 2, 2) = S(2) * 0 * R(2) = S(2) * R(2)
 \end{aligned}$$

上述した各分割部分データD(i, j)を生成するための定義式は、図3にも図示されている。

【0055】

図3は、上述したように16ビットの元データSを8ビットの処理単位ビット

長に基づいて分割数 $n=3$ で3分割する場合の各データと定義式および各分割部分データから元データを復元する場合の計算式などを示す表である。

【0056】

ここで、上述した定義式により分割データ $D(1), D(2), D(3)$ および各分割部分データ $D(1,1), D(1,2), D(2,1), D(2,2), D(3,1), D(3,2)$ を生成する過程と定義式の一般形について説明する。

【0057】

まず、第1の分割データ $D(1)$ に対しては、第1の分割部分データ $D(1,1)$ は、上述した定義式 $S(1)*R(1)*R(2)$ で定義され、第2の分割部分データ $D(1,2)$ は定義式 $S(2)*R(1)*R(2)$ で定義される。なお、この定義式の一般形は、 $D(1,j)$ に対しては $S(j)*R(j)*R(j+1)$ であり、 $D(1,j+1)$ に対して $S(j+1)*R(j)*R(j+1)$ である (j は奇数とする)。定義式に従って計算すると、 $D(1,1)$ は00110110, $D(1,2)$ は10110011となるので、 $D(1)$ は00110110 10110011である。なお、定義式の一般形は、図4にまとめて示されている。

【0058】

また、第2の分割データ $D(2)$ に対しては、 $D(2,1)$ は $S(1)*R(1)$ で定義され、 $D(2,2)$ は $S(2)*R(2)$ で定義される。この定義式の一般形は、 $D(2,j)$ に対しては $S(j)*R(j)$ であり、 $D(2,j+1)$ に対しては $S(j+1)*R(j+1)$ である (j は奇数とする)。定義式に従って計算すると、 $D(2,1)$ は00000011, $D(2,2)$ は00000010となるので、 $D(2)$ は00000011 00000010である。

【0059】

更に第3の分割データ $D(3)$ に対しては、 $D(3,1)$ は $R(1)$ で定義され、 $D(3,2)$ は $R(2)$ で定義される。この定義式の一般形は、 $D(3,j)$ に対しては $R(j)$ であり、 $D(3,j+1)$ に対しては $R(j+1)$ である (j は奇数とする)。定義式に従って計算すると、 $D(3,1)$ は10110001, $D(3,2)$ は00110101となるので、 $D(3)$ は10110001 00110101である。

【0060】

上記説明は、 $S, R, D(1), D(2), D(3)$ の長さを16ビットとしたが、データの先頭から上記分割処理を繰り返すことにより、どのような長さの元データ S からでも

分割データD(1),D(2),D(3)を生成することができる。また、処理単位ビット長bは任意にとることができる、元データSの先頭から順にb×2の長さ毎に上記分割処理を繰り返すことにより任意の長さの元データ、具体的には処理単位ビット長b×2の整数倍の長さの元データに対して適用することができる。なお、元データSの長さが処理単位ビット長b×2の整数倍でない場合は、例えば、データ末尾の部分を0で埋めるなどして元データSの長さを処理単位ビット長b×2の整数倍に合わせるにより上述した本実施形態の分割処理を適用することができる。

【0061】

次に、図3の右側に示す表を参照して、分割データから元データを復元する処理について説明する。

【0062】

まず、利用者は端末5からネットワーク3を介して分割装置1にアクセスし、分割装置1のデータ送受信手段17を介して元データSの復元を要求する。分割装置1は、この元データSの復元要求を受け取ると、この元データSに対応する分割データD(1),D(2),D(3)が保管サーバ7a, 7b, 7cに保管されていることを記憶しているので、ネットワーク3を介して保管サーバ7a, 7b, 7cから分割データD(1),D(2),D(3)を取得し、この取得した分割データD(1),D(2),D(3)から次に示すように元データSを復元する。

【0063】

まず、分割部分データD(2,1),D(3,1)から第1の元部分データS(1)を次のように生成することができる。

【0064】

$$\begin{aligned} D(2,1)*D(3,1) &= (S(1)*R(1))*R(1) \\ &= S(1)*(R(1)*R(1)) \\ &= S(1)*0 \\ &= S(1) \end{aligned}$$

具体的に計算すると、D(2,1)は00000011, D(3,1)は10110001なので、S(1)は10110010となる。

【0065】

また、別の分割部分データから次のように第2の元部分データS(2)を生成することができる。

【0 0 6 6】

$$\begin{aligned} D(2,2)*D(3,2) &= (S(2)*R(2))*R(2) \\ &= S(2)*(R(2)*R(2)) \\ &= S(2)*0 \\ &= S(2) \end{aligned}$$

具体的に計算すると、D(2,2)は00000010, D(3,2)は00110101なので、S(2)は00110111となる。

【0 0 6 7】

一般に、jを奇数として、

$$\begin{aligned} D(2,j)*D(3,j) &= (S(j)*R(j))*R(j) \\ &= S(j)*(R(j)*R(j)) \\ &= S(j)*0 \\ &= S(j) \end{aligned}$$

であるから、D(2,j)*D(3,j)を計算すれば、S(j)が求まる。

【0 0 6 8】

また、一般に、jを奇数として、

【数1 1】

$$\begin{aligned} D(2,j+1)*D(3,j+1) &= (S(j+1)*R(j+1))*R(j+1) \\ &= S(j+1)*(R(j+1)*R(j+1)) \\ &= S(j+1)*0 \\ &= S(j+1) \end{aligned}$$

であるから、D(2,j+1)*D(3,j+1)を計算すれば、S(j+1)が求まる。

【0 0 6 9】

次に、D(1),D(3)を取得してSを復元する場合には、次のようになる。

【0 0 7 0】

【数1 2】

$$D(1,1)*D(3,1)*D(3,2) = (S(1)*R(1)*R(2))*R(1)*R(2)$$

$$=S(1)*(R(1)*R(1))*(R(2)*R(2))$$

$$=S(1)*0*0$$

$$=S(1)$$

であるから、 $D(1,1)*D(3,1)*D(3,2)$ を計算すれば、 $S(1)$ が求まる。具体的に計算すると、 $D(1,1)$ は00110110, $D(3,1)$ は10110001, $D(3,2)$ は00110101なので、 $S(1)$ は10110010となる。

【0071】

また同様に、

【数13】

$$D(1,2)*D(3,1)*D(3,2)=(S(2)*R(1)*R(2))*R(1)*R(2)$$

$$=S(2)*(R(1)*R(1))*(R(2)*R(2))$$

$$=S(2)*0*0$$

$$=S(2)$$

であるから、 $D(1,2)*D(3,1)*D(3,2)$ を計算すれば、 $S(2)$ が求まる。具体的に計算すると、 $D(1,2)$ は10110011, $D(3,1)$ は10110001, $D(3,2)$ は00110101なので、 $S(2)$ は00110111となる。

【0072】

一般に、 j を奇数として、

【数14】

$$D(1,j)*D(3,j)*D(3,j+1)=(S(j)*R(j)*R(j+1))*R(j)*R(j+1)$$

$$=S(j)*(R(j)*R(j))*(R(j+1)*R(j+1))$$

$$=S(j)*0*0$$

$$=S(j)$$

であるから、 $D(1,j)*D(3,j)*D(3,j+1)$ を計算すれば、 $S(j)$ が求まる。

【0073】

また、一般に、 j を奇数として、

【数15】

$$D(1,j+1)*D(3,j)*D(3,j+1)=(S(j+1)*R(j)*R(j+1))*R(j)*R(j+1)$$

$$=S(j+1)*(R(j)*R(j))*(R(j+1)*R(j+1))$$

$$=S(j+1)*0*0$$

$$=S(j+1)$$

であるから、 $D(1, j+1)*D(3, j)*D(3, j+1)$ を計算すれば、 $S(j+1)$ が求まる。

【0 0 7 4】

次に、 $D(1), D(2)$ を取得して S を復元する場合には、次のようになる。

【0 0 7 5】

【数 1 6】

$$\begin{aligned} D(1, 1)*D(2, 1) &= (S(1)*R(1)*R(2))*(S(1)*R(1)) \\ &= (S(1)*S(1))*(R(1)*R(1))*R(2) \\ &= 0*0*R(2) \\ &= R(2) \end{aligned}$$

であるから、 $D(1, 1)*D(2, 1)$ を計算すれば、 $R(2)$ が求まる。具体的に計算すると、 $D(1, 1)$ は00110110, $D(2, 1)$ は00000011なので、 $R(2)$ は00110101となる。

【0 0 7 6】

また同様に、

【数 1 7】

$$\begin{aligned} D(1, 2)*D(2, 2) &= (S(2)*R(1)*R(2))*(S(2)*R(2)) \\ &= (S(2)*S(2))*R(1)*(R(2)*R(2)) \\ &= 0*R(1)*0 \\ &= R(1) \end{aligned}$$

であるから、 $D(1, 2)*D(2, 2)$ を計算すれば、 $R(1)$ が求まる。具体的に計算すると、 $D(1, 2)$ は10110011, $D(2, 2)$ は00000010なので、 $R(1)$ は10110001となる。

【0 0 7 7】

この $R(1), R(2)$ を使用して $S(1), S(2)$ を求める。

【0 0 7 8】

$$\begin{aligned} D(2, 1)*R(1) &= (S(1)*R(1))*R(1) \\ &= S(1)*(R(1)*R(1)) \\ &= S(1)*0 \\ &= S(1) \end{aligned}$$

であるから、 $D(2,1)*R(1)$ を計算すれば、 $S(1)$ が求まる。具体的に計算すると、 $D(2,1)$ は00000011, $R(1)$ は10110001なので、 $S(1)$ は10110010となる。

【0 0 7 9】

また同様に、

$$\begin{aligned} D(2,2)*R(2) &= (S(2)*R(2))*R(2) \\ &= S(2)*(R(2)*R(2)) \\ &= S(2)*0 \\ &= S(2) \end{aligned}$$

であるから $D(2,2)*R(2)$ を計算すれば $S(2)$ が求まる。具体的に計算すると $D(2,2)$ は00000010, $R(2)$ は00110101なので、 $S(2)$ は00110111となる。

【0 0 8 0】

一般に、 j を奇数として、

【数 1 8】

$$\begin{aligned} D(1,j)*D(2,j) &= (S(j)*R(j)*R(j+1))*(S(j)*R(j)) \\ &= (S(j)*S(j))*(R(j)*R(j))*R(j+1) \\ &= 0*0*R(j+1) \\ &= R(j+1) \end{aligned}$$

であるから $D(1,j)*D(2,j)$ を計算すれば $R(j+1)$ が求まる。

【0 0 8 1】

また同様に、

【数 1 9】

$$\begin{aligned} D(1,j+1)*D(2,j+1) &= (S(j+1)*R(j)*R(j+1))*(S(j+1)*R(j+1)) \\ &= (S(j+1)*S(j+1))*R(j)*(R(j+1)*R(j+1)) \\ &= 0*R(j)*0 \\ &= R(j) \end{aligned}$$

であるから $D(1,j+1)*D(2,j+1)$ を計算すれば $R(j)$ が求まる。

【0 0 8 2】

この $R(j), R(j+1)$ を使用して $S(j), S(j+1)$ を求める。

【0 0 8 3】

$$\begin{aligned} D(2, j) * R(j) &= (S(j) * R(j)) * R(j) \\ &= S(j) * (R(j) * R(j)) \\ &= S(j) * 0 \\ &= S(j) \end{aligned}$$

であるから $D(2, j) * R(j)$ を計算すれば $S(j)$ が求まる。

【0 0 8 4】

また同様に、

$$\begin{aligned} D(2, j+1) * R(j+1) &= (S(j+1) * R(j+1)) * R(j+1) \\ &= S(j+1) * (R(j+1) * R(j+1)) \\ &= S(j+1) * 0 \\ &= S(j+1) \end{aligned}$$

であるから $D(2, j+1) * R(j+1)$ を計算すれば $S(j+1)$ が求まる。

【0 0 8 5】

上述したように、元データの先頭から処理単位ビット長 b に基づいて分割処理を繰り返し行って、分割データを生成した場合には、3つの分割データ $D(1), D(2), D(3)$ のすべてを用いなくても、3つの分割データのうち、2つの分割データを用いて上述したように元データを復元することができる。

【0 0 8 6】

本発明の他の実施形態として、乱数 R のビット長を元データ S のビット長よりも短いものを使用して、元データの分割処理を行うことができる。

【0 0 8 7】

すなわち、上述した乱数 R は $S, D(1), D(2), D(3)$ と同じビット長のデータとしたが、乱数 R を元データ S のビット長より短いものとし、分割データ $D(1), D(2), D(3)$ の生成にこの短いビット長の乱数 R を繰り返し用いるものである。

【0 0 8 8】

なお、分割データ $D(3)$ は乱数 R のみから生成されるので、分割データ $D(3)$ は乱数 R を繰り返して保管しておく必要はない。例えば、元データ S のビット長を 1 6 0 0 ビット (200 バイト) としたとき、乱数 R は任意にとった 1 6 0 ビット (20 バイト) のデータの繰り返しとする。つまり、 $R(1) \sim R(20)$ はランダムに生成し、 R

(21)~R(200)は $R(21)=R(1)$, $R(22)=R(2)$, ..., $R(40)=R(20)$, $R(41)=R(1)$, $R(42)=R(2)$, ..., $R(60)=R(20)$, $R(61)=R(1)$, $R(62)=R(2)$, ..., $R(80)=R(20)$,, $R(181)=R(1)$, $R(182)=R(2)$, ..., $R(200)=R(20)$ とする。

【0089】

先の説明では、分割部分データ $D(3, j)$ を乱数部分データ $R(j)$ と定義して $D(3)$ を生成しているが、 $D(3, 20)$ まで保管すれば十分である。つまり、 $D(3)$ の長さは $D(1)$, $D(2)$ の10分の1となる。従って、保管すべきデータの総量は先の実施形態では元データ S の3倍であるが、この実施形態では2.1倍とすることができる。乱数 R における繰り返し部分のデータの長さは、短すぎると、 $D(1)$ または $D(2)$ のみから R が解読されてしまうことも考えられるため、適切な長さを選択することが望ましい。

【0090】

上述した各実施形態は、元データを3つに分割し、そのうち2つから元データが復元可能となるものであったが、分割数 n を3より大きくとって、 n 個より少ない個数の分割データから元データを復元することができることは勿論のことである。

【0091】

その1つの応用例として、元データを4つの分割データに分割する分割数 $n=4$ の場合の分割処理について図5に示すフローチャートおよび図6に示す定義式の一般形などを参照して説明する。

【0092】

まず、利用者は端末5から分割装置1にアクセスして元データ S を送信し、分割装置1ではデータ送受信手段17が端末5からの元データ S を受信し、分割装置1に供給する(ステップS301)。それから、利用者は端末5から分割数 n として4を分割装置1に指示する(ステップS303)。この分割数 n は分割装置1において予め定められた値を用いてもよい。また、処理単位ビット長 b が一例として8ビットと決定される(ステップS305)。次に、元データ S のビット長が 8×3 の整数倍であるか否かを判定し、整数倍でない場合には、元データ S の末尾を0で埋める(ステップS307)。また、整数倍を意味する変数 m を0

に設定する（ステップS309）。

【0093】

次に、元データSの $8 \times 3 \times m + 1$ ビット目から 8×3 ビット分のデータが存在するかどうか判定される（ステップS311）。なお、本例では、元データSが $8 \times 3 = 24$ ビット長のデータの場合について説明している。

【0094】

ステップS311の判定の結果、本例の元データSでは $8 \times 3 = 24$ ビットのデータであり、変数 $m=1$ の場合に相当する $8 \times 3 \times m (=1) + 1$ ビットに相当する25ビット以降のデータは存在しないので、ステップS311からステップS321に進むことになるが、今の場合は、変数 m は0であるので、元データSの $8 \times 3 \times m + 1$ ビット目は、 $8 \times 3 \times 0 + 1 = 1$ となり、元データSの24ビットの1ビット目から 8×3 ビット分にデータが存在するため、ステップS313に進む。

【0095】

ステップS313では、変数 j を1から3（=分割数 $n-1$ ）まで変えて、元データSの $8 \times (3 \times m + j - 1) + 1$ ビット目から8ビット分（=処理単位ビット長）のデータを元部分データ $S(3 \times m + j)$ に設定し、これにより元データSを処理単位ビット長で区分けした3個の元部分データ $S(1), S(2), S(3)$ が生成される。

【0096】

次に、変数 j を1から3まで変えて、乱数部分データ $R(3 \times m + j)$ に乱数発生手段15から発生する8ビットの長さの乱数を設定し、これにより乱数 R を処理単位ビット長で区分けした3個の乱数部分データ $R(1), R(2), R(3)$ が生成される（ステップS315）。

【0097】

次に、ステップS317において、乱数 i を1から4（=分割数 n ）まで変えるとともに、更に各変数 i において変数 j を1から3（=分割数 $n-1$ ）まで変えながら、ステップS317に示す分割データを生成するための定義式により複数の分割データ $D(i)$ の各々を構成する各分割部分データ $D(i, 3 \times m + j)$ を生成する。この結果、次に示すような分割データ D が生成される。

【0098】

【数 20】

分割データD

=4個の分割データ $D(i)=D(1), D(2), D(3), D(4)$

第1の分割データD(1)

=3個の分割部分データ $D(1, j)=D(1, 1), D(1, 2), D(1, 3)$

第2の分割データD(2)

=3個の分割部分データ $D(2, j)=D(2, 1), D(2, 2), D(2, 3)$

第3の分割データD(3)

=3個の分割部分データ $D(3, j)=D(3, 1), D(3, 2), D(3, 3)$

第4の分割データD(4)

=3個の分割部分データ $D(4, j)=D(4, 1), D(4, 2), D(4, 3)$

なお、各分割部分データ $D(i, j)$ を生成するためのステップS317に示す定義式は、本例のように分割数 $n=4$ の場合には、具体的には図6に示す表に記載されているものとなる。図6に示す表から、分割部分データ $D(1, 1)$ を生成するための定義式は $S(1)*R(1)*R(2)*R(3)$ であり、 $D(1, 2)$ の定義式は $S(2)*R(1)*R(2)*R(3)$ であり、 $D(1, 3)$ の定義式は $S(3)*R(1)*R(2)*R(3)$ であり、 $D(2, 1)$ の定義式は $S(1)*R(1)*R(2)$ であり、 $D(2, 2)$ の定義式は $S(2)*R(2)*R(3)$ であり、 $D(2, 3)$ の定義式は $S(3)*R(1)*R(3)$ であり、 $D(3, 1)$ の定義式は $S(1)*R(1)$ であり、 $D(3, 2)$ の定義式は $S(2)*R(2)$ であり、 $D(3, 3)$ の定義式は $S(3)*R(3)$ であり、 $D(4, 1)$ の定義式は $R(1)$ であり、 $D(4, 2)$ の定義式は $R(2)$ であり、 $D(4, 3)$ の定義式は $R(3)$ である。また、図6に示す表には $m>0$ の場合の任意の整数についての一般的な定義式も記載されている。

【0099】

このように変数 $m=0$ の場合について分割データDを生成した後、次に変数 m を1増やし（ステップS319）、ステップS311に戻り、変数 $m=1$ に該当する元データSの25ビット以降について同様の分割処理を行おうとするが、本例の元データSは24ビットであり、25ビット以降のデータは存在しないので、ステップS311からステップS321に進み、上述したように生成した分割データ $D(1), D(2), D(3), D(4)$ を分割装置1のデータ送受信手段17からネットワーク3を介して保管サーバ7にそれぞれ送信し、各保管サーバ7に保管し、分割処理を

終了する。図1では保管サーバは3個であるが、分割数に応じて保管サーバを増やし、各分割データを異なる保管サーバに保管することが望ましい。

【0100】

ここで、上述した図5のフローチャートのステップS317における定義式による分割データの生成処理、具体的には分割数 $n=4$ の場合の分割データの生成処理について詳しく説明する。

【0101】

まず、ステップS317に示す定義式から各分割データ $D(i)=D(1) \sim D(4)$ の各々を構成する各分割部分データ $D(i, 3 \times m + j)$ は、次のようになる。

【0102】

【数21】

$$D(1, 3 \times m + 1) = S(3 \times m + 1) * Q(1, 1, 1) * Q(1, 1, 2) * Q(1, 1, 3)$$

$$D(1, 3 \times m + 2) = S(3 \times m + 2) * Q(2, 1, 1) * Q(2, 1, 2) * Q(2, 1, 3)$$

$$D(1, 3 \times m + 3) = S(3 \times m + 3) * Q(3, 1, 1) * Q(3, 1, 2) * Q(3, 1, 3)$$

$$D(2, 3 \times m + 1) = S(3 \times m + 1) * Q(1, 2, 1) * Q(1, 2, 2) * Q(1, 2, 3)$$

$$D(2, 3 \times m + 2) = S(3 \times m + 2) * Q(2, 2, 1) * Q(2, 2, 2) * Q(2, 2, 3)$$

$$D(2, 3 \times m + 3) = S(3 \times m + 3) * Q(3, 2, 1) * Q(3, 2, 2) * Q(3, 2, 3)$$

$$D(3, 3 \times m + 1) = S(3 \times m + 1) * Q(1, 3, 1) * Q(1, 3, 2) * Q(1, 3, 3)$$

$$D(3, 3 \times m + 2) = S(3 \times m + 2) * Q(2, 3, 1) * Q(2, 3, 2) * Q(2, 3, 3)$$

$$D(3, 3 \times m + 3) = S(3 \times m + 3) * Q(3, 3, 1) * Q(3, 3, 2) * Q(3, 3, 3)$$

$$D(4, 3 \times m + 1) = R(3 \times m + 1)$$

$$D(4, 3 \times m + 2) = R(3 \times m + 2)$$

$$D(4, 3 \times m + 3) = R(3 \times m + 3)$$

次に、 $Q(j, i, k)$ を具体的に求める。これは $c(j, i, k)$ を 3×3 行列である $U[3, 3] \times (P[3, 3])^{(j-1)}$ の i 行 k 列の値としたとき下記のように定義される。

【0103】

$$c(j, i, k) = 1 \text{ のとき } Q(j, i, k) = R(3 \times m + k)$$

$$c(j, i, k) = 0 \text{ のとき } Q(j, i, k) = 0$$

$j=1$ のときは

【数 2 2】

$$U[3, 3] \times (P[3, 3])^{-(j-1)} = U[3, 3] \times (P[3, 3])^{-0} = U[3, 3]$$

$$= \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

j=2のときは

【数 2 3】

$$U[3, 3] \times (P[3, 3])^{-(j-1)} = U[3, 3] \times (P[3, 3])^{-1} = U[3, 3] \times (P[3, 3])$$

$$= \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

j=3のときは

【数 2 4】

$$U[3, 3] \times (P[3, 3])^{-(j-1)} = U[3, 3] \times (P[3, 3])^{-2} = U[3, 3] \times (P[3, 3]) \times (P[3, 3])$$

$$= \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

これを用いると、各分割部分データは次のような定義式により生成される。

【0 1 0 4】

【数 2 5】

$$\begin{aligned} D(1, 3 \times m + 1) &= S(3 \times m + 1) * Q(1, 1, 1) * Q(1, 1, 2) * Q(1, 1, 3) \\ &= S(3 \times m + 1) * R(3 \times m + 1) * R(3 \times m + 2) * R(3 \times m + 3) \end{aligned}$$

$$\begin{aligned} D(1, 3 \times m + 2) &= S(3 \times m + 2) * Q(2, 1, 1) * Q(2, 1, 2) * Q(2, 1, 3) \\ &= S(3 \times m + 2) * R(3 \times m + 1) * R(3 \times m + 2) * R(3 \times m + 3) \end{aligned}$$

$$\begin{aligned} D(1, 3 \times m + 3) &= S(3 \times m + 3) * Q(3, 1, 1) * Q(3, 1, 2) * Q(3, 1, 3) \\ &= S(3 \times m + 3) * R(3 \times m + 1) * R(3 \times m + 2) * R(3 \times m + 3) \end{aligned}$$

$$\begin{aligned} D(2, 3 \times m + 1) &= S(3 \times m + 1) * Q(1, 2, 1) * Q(1, 2, 2) * Q(1, 2, 3) \\ &= S(3 \times m + 1) * R(3 \times m + 1) * R(3 \times m + 2) \end{aligned}$$

$$\begin{aligned} D(2, 3 \times m + 2) &= S(3 \times m + 2) * Q(2, 2, 1) * Q(2, 2, 2) * Q(2, 2, 3) \\ &= S(3 \times m + 2) * R(3 \times m + 2) * R(3 \times m + 3) \end{aligned}$$

$$\begin{aligned} D(2, 3 \times m + 3) &= S(3 \times m + 3) * Q(3, 2, 1) * Q(3, 2, 2) * Q(3, 2, 3) \\ &= S(3 \times m + 3) * R(3 \times m + 1) * R(3 \times m + 3) \end{aligned}$$

$$\begin{aligned} D(3, 3 \times m + 1) &= S(3 \times m + 1) * Q(1, 3, 1) * Q(1, 3, 2) * Q(1, 3, 3) \\ &= S(3 \times m + 1) * R(3 \times m + 1) \end{aligned}$$

$$\begin{aligned} D(3, 3 \times m + 2) &= S(3 \times m + 2) * Q(2, 3, 1) * Q(2, 3, 2) * Q(2, 3, 3) \\ &= S(3 \times m + 2) * R(3 \times m + 2) \end{aligned}$$

$$\begin{aligned} D(3, 3 \times m + 3) &= S(3 \times m + 3) * Q(3, 3, 1) * Q(3, 3, 2) * Q(3, 3, 3) \\ &= S(3 \times m + 3) * R(3 \times m + 3) \end{aligned}$$

$$D(4, 3 \times m + 1) = R(3 \times m + 1)$$

$$D(4, 3 \times m + 2) = R(3 \times m + 2)$$

$$D(4, 3 \times m + 3) = R(3 \times m + 3)$$

ここで、上述したように図2のステップS217や図5のステップS317で示した定義式に基づいて元データを分割する分割規則について一般的な表現で記載する。

【0105】

まず、元データ、乱数、分割データ、分割数および処理単位ビット長をそれぞれS, R, D, nおよびbで表すとともに、複数n個のうちの1つを表わす変数としてi(=1~n)およびj(=1~n-1)を用いて複数(n-1)個の元部分データ、複数(n-1)個の乱数部分データ、複数(n)個の分割データおよび各分割データの複数(n-1)個の分割部分データのそれぞれのうちの1つをそれぞれS(j), R(j), D(i)およびD(i, j)で表わす。

【0106】

それから、上記変数jを1からn-1まで変えて、各元部分データS(j)を元データSのb×(j-1)+1ビット目からbビット分のデータとして作成する。次に、U[n, n]をn×n行列である上三角行列とし、P[n, n]をn×n行列である回転行列としたとき、

$c(j, i, k)$ を $(n-1) \times (n-1)$ 行列である $U[n-1, n-1] \times (P[n-1, n-1])^{(j-1)}$ の i 行 k 列の値と定義する。そして、 $c(j, i, k)=1$ のとき、 $Q(j, i, k)=R(k)$ 、 $c(j, i, k)=0$ のとき、 $Q(j, i, k)=0$ と定義したとき、変数 i を 1 から n まで変えながら、各変数 i において変数 j を 1 から $n-1$ まで変えた場合において、

$i < n$ のとき、各分割部分データ $D(i, j)$ を

【数 2 6】

$$D(i, j) = S(j) * \left(\prod_{k=1}^{n-1} Q(j, i, k) \right)$$

と設定し、また $i=n$ のとき、各分割部分データ $D(i, j)$ を

$$D(i, j) = R(j)$$

と設定する。上記処理を元データ S の先頭から末尾まで繰り返し行うことにより元データ S から分割数 n の分割データを生成することができる。

【0107】

次に、上述したように元データ S を 4 分割して生成された分割データ $D(1), D(2), D(3), D(4)$ から元データ S を復元する処理について図 6 を参照して説明する。なお、図 6 に示す 4 分割の場合には、変数 j を $3 \times m + 1$ ($m \geq 0$ である任意の整数) として、同図に示す一般的な定義式から次に示すように元データ S を生成することができる。

【0108】

まず、分割データ $D(1), D(2)$ から元データ S を求める場合について説明する。

【0109】

【数 2 7】

$$\begin{aligned} D(1, j) * D(2, j) &= (S(j) * R(j) * R(j+1) * R(j+2)) * ((S(j) * R(j) * R(j+1)) \\ &= (S(j) * S(j)) * (R(j) * R(j)) * (R(j+1) * R(j+1)) * R(j+2) \\ &= 0 * 0 * 0 * R(j+2) \\ &= R(j+2) \end{aligned}$$

従って、 $D(1, j) * D(2, j)$ を計算すれば、乱数 $R(j+2)$ が求まり、同様に $D(1, j+1) * D(2, j+1)$ を計算すれば、乱数 $R(j)$ が求まり、同様に $D(1, j+2) * D(2, j+2)$ を計算すれば、乱数 $R(j+1)$ が求まり、これらの得られた乱数 $R(j), R(j+1), R(j+2)$ を用いれ

ば、

【数 2 8】

$$\begin{aligned}
 & D(1, j) * R(j) * R(j+1) * R(j+2) \\
 &= (S(j) * R(j) * R(j+1) * R(j+2)) * (R(j) * R(j+1) * R(j+2)) \\
 &= S(j) * (R(j) * R(j+1)) * (R(j+1) * R(j+1)) * (R(j+2) * R(j+2)) \\
 &= S(j) * 0 * 0 * 0 \\
 &= S(j)
 \end{aligned}$$

であるから、 $D(1, j) * R(j) * R(j+1) * R(j+2)$ を計算して、 $S(j)$ を求めてもよいし、 $D(2, j) * R(j) * R(j+1)$ から $S(j)$ を求めることもできる。

【0 1 1 0】

同様に、 $D(1, j+1) * R(j) * R(j+1) * R(j+2)$ または $D(2, j+1) * R(j+1) * R(j+2)$ を計算して $S(j+1)$ を求めることができ、また同様に $D(1, j+2) * R(j) * R(j+1) * R(j+2)$ または $D(2, j+2) * R(j) * R(j+2)$ を計算して $S(j+2)$ を求めることができる。

【0 1 1 1】

更に、上述したと同様に、 $D(2)$ と $D(3)$ から S を求めることができる。

【0 1 1 2】

具体的には、まず $R(j), R(j+1), R(j+2)$ を求めてから、 $D(2, j), D(2, j+1), D(2, j+2)$ または $D(3, j), D(3, j+1), D(3, j+2)$ と $R(j), R(j+1), R(j+2)$ のXOR演算により $S(j), S(j+1), S(j+2)$ を求めることができる。

【0 1 1 3】

また更に、 $D(1)$ と $D(4)$ または $D(2)$ と $D(4)$ または $D(3)$ と $D(4)$ から S を求めることができる。

【0 1 1 4】

$D(4)$ は R をそのものから定義したものであるから、計算することなく $D(4)$ から $R(j), R(j+1), R(j+2)$ を取得することができ、例えば、 $D(1, j), D(1, j+1), D(1, j+2)$ と $R(j), R(j+1), R(j+2)$ のXOR演算により $S(j), S(j+1), S(j+2)$ を求めることができる。

【0 1 1 5】

上述したように、演算回数の差が1である任意の2つの分割データ $D(1)$ と $D(2)$

、または、 $D(2)$ と $D(3)$ 、または、 $D(4)$ と任意の1つの分割データ $D(1)$ または $D(2)$ または $D(3)$ から S が復元可能である。すなわち、4つの分割データの中から任意に3つの分割データを取得すれば、その中には必ず上述したいずれかのケースが含まれるため、4つのうち任意の3つの分割データから元データを復元可能である。

【0116】

図7は、5分割の場合の分割データと定義式を示す表である。この5分割の場合は、 j を $4 \times m + 1$ (m は $m \geq 0$ である任意の整数) として、分割データの定義式から、上述した4分割の場合の復元処理と同様のことが言える。従って、演算回数の差が1である任意の2つの分割データ $D(1)$ と $D(2)$ 、または、 $D(2)$ と $D(3)$ 、または、 $D(3)$ と $D(4)$ 、または、 $D(5)$ と任意の1つの分割データ $D(1)$ または $D(2)$ または $D(3)$ または $D(4)$ から元データ S が復元可能である。そして、5つの分割データの中から任意に3つの分割データを取得すれば、その中には必ずこのいずれかのケースが含まれるため、5つのうち任意の3つから復元可能であるといえる。

【0117】

また、分割数 n を5より大きくとった場合も同様にして分割データを構成すれば、 n が奇数である場合は $(n+1)/2$ 個、 n が偶数である場合は $(n/2)+1$ 個の分割データから元データを復元することができる。この個数は、 n 個の分割データがあったときに、隣り合ったものを選択せず、かつ、 n 個目の分割データを選択しないような最大個数に1を加えたものである。つまり、前記最大個数に1を加えれば演算回数の差が1である2つの分割データまたは n 個目の分割データとその他のデータを必ず含むこととなるため、復元に必要な個数が前記のとおりといえる。

【0118】

次に、図8に示すフローチャートを参照して、分割数が n で、処理単位ビット長が b である場合の一般的な分割処理について説明する。

【0119】

まず、利用者は端末5から分割装置1にアクセスして元データ S を送信し、分割装置1ではデータ送受信手段17が端末5からの元データ S を受信し、分割装置1に供給する(ステップS401)。また、利用者は端末5から分割数 n ($n \geq 3$)

である任意の整数)を分割装置1に指示する(ステップS403)。この分割数 n は分割装置1において予め定められた値を用いてもよい。処理単位ビット長 b を決定する(ステップS405)。なお、 b は0より大きい任意の整数である。次に、元データ S のビット長が $b \times (n-1)$ の整数倍であるか否かを判定し、整数倍でない場合には、元データ S の末尾を0で埋める(ステップS407)。また、整数倍を意味する変数 m を0に設定する(ステップS409)。

【0120】

次に、元データ S の $b \times (n-1) \times m+1$ ビット目から $b \times (n-1)$ ビット分のデータが存在するか否かが判定される(ステップS411)。この判定の結果、データが存在しない場合は、ステップS421に進むことになるが、今の場合は、ステップS409で変数 m は0に設定された場合であるので、データが存在するため、ステップS413に進む。

【0121】

ステップS413では、変数 j を1から $n-1$ まで変えて、元データ S の $b \times (n-1) \times m+j-1+1$ ビット目から b ビット分のデータを元部分データ $S((n-1) \times m+j)$ に設定する処理を繰り返し、これにより元データ S を処理単位ビット長 b で分けした $(n-1)$ 個の元部分データ $S(1), S(2), \dots, S(n-1)$ が生成される。

【0122】

次に、変数 j を1から $n-1$ まで変えて、乱数部分データ $R((n-1) \times m+j)$ に乱数発生手段15から発生する処理単位ビット長 b の乱数を設定し、これにより乱数 R を処理単位ビット長 b で分けした $n-1$ 個の乱数部分データ $R(1), R(2), \dots, R(n-1)$ が生成される(ステップS415)。

【0123】

次に、ステップS417において、変数 i を1から n まで変えるとともに、更に各変数 i において変数 j を1から $n-1$ まで変えながら、ステップS417に示す分割データを生成するための定義式により複数の分割データ $D(i)$ の各々を構成する各分割部分データ $D(i, (n-1) \times m+j)$ を生成する。この結果、次に示すような分割データ D が生成される。

【0124】

【数 2 9】

分割データD

=n個の分割データ $D(i)=D(1), D(2), \dots, D(n)$

第 1 の分割データD(1)

=n-1個の分割部分データ $D(1, j)=D(1, 1), D(1, 2), \dots, D(1, n-1)$

第 2 の分割データD(2)

=n-1個の分割部分データ $D(2, j)=D(2, 1), D(2, 2), \dots, D(2, n-1)$

...

...

第nの分割データD(n)

=n-1個の分割部分データ $D(n, j)=D(n, 1), D(n, 2), \dots, D(n, n-1)$

このように変数 $m=0$ の場合について分割データDを生成した後、次に変数 m を 1 増やし（ステップ S 4 1 9）、ステップ S 4 1 1に戻り、変数 $m=1$ に該当する元データSの $b \times (n-1)$ ビット以降について同様の分割処理を行う。最後にステップ S 4 1 1の判定の結果、元データSにデータがなくなった場合、ステップ S 4 1 1からステップ S 4 2 1に進み、上述したように生成した分割データDを分割装置 1 のデータ送受信手段 1 7 からネットワーク 3 を介して保管サーバ 7 にそれぞれ送信し、各保管サーバ 7 に保管し、分割処理を終了する。図 1 では保管サーバは 3 個であるが、分割数に応じて保管サーバを増やし、各分割データを異なる保管サーバに保管することが望ましい。

【0 1 2 5】

次に、図 9 に示すフローチャートを参照して、分割数 n が 2 の場合の分割処理について説明する。すなわち、上述した各実施形態は図 8 のフローチャートのステップ S 4 0 3 に示したように分割数 n が 3 以上($n \geq 3$)の場合についてのものであるので、図 9 を用いて分割数 n が 2 の場合について説明する。

【0 1 2 6】

まず、利用者は端末 5 から分割装置 1 にアクセスして元データSを分割装置 1 に供給する（ステップ S 5 0 1）。また、利用者は端末 5 から分割数 n として 2 を分割装置 1 に指示する（ステップ S 5 0 3）。この分割数 n は分割装置 1 にお

いて予め定められた値を用いてもよい。それから処理単位ビット長 b として8ビットを決定する(ステップS505)。次に、元データSのビット長が8の整数倍であるか否かを判定し、整数倍でない場合には、元データSの末尾を0で埋める(ステップS507)。また、整数倍を意味する変数 m を0に設定する(ステップS509)。

【0127】

次に、元データSの $8 \times m + 1$ ビット目から8ビット分のデータが存在するか否かが判定される(ステップS511)。この判定の結果、データが存在しない場合は、ステップS521に進むことになるが、今の場合は、変数 m は0に設定されているので、データが存在するため、ステップS513に進む。

【0128】

ステップS513では、元データSの $8 \times m + 1$ ビット目から8ビット分のデータを元部分データS($m+1$)に設定し、これにより元部分データS(1)が生成される。

【0129】

次に、乱数部分データR($m+1$)に乱数発生手段15から発生する8ビットの乱数を設定し、これにより乱数部分データR(1)が生成される(ステップS515)。

【0130】

次に、ステップS517において、同ステップに示す定義式により分割データDの各々を構成する各分割データD(1, $m+1$), D(2, $m+1$)が生成される。

【0131】

このように変数 $m=0$ の場合について分割データDを生成した後、次に変数 m を1増やし(ステップS519)、ステップS511に戻り、変数 $m=1$ に該当する元データSの8ビット以降について同様の分割処理を行う。最後にステップS511の判定の結果、元データSにデータがなくなった場合、ステップS511からステップS521に進み、上述したように生成した分割データD(1)からD(2)を分割装置1のデータ送受信手段17からネットワーク3を介して保管サーバ7にそれぞれ送信し、各保管サーバ7に保管し、分割処理を終了する。図1では保管サーバは3個であるが、このうち2個の保管サーバに各分割データを保管すればよい。

【0132】

ここにおいて、上述した図9のフローチャートのステップS517における定義式による分割データの生成処理、具体的には分割数 $n=2$ の場合の分割データの生成処理について詳しく説明する。

【0133】

変数 $m=0$ の場合には、ステップS517に示す定義式から各分割データ $D(1,1)$, $D(2,1)$ は、次のようになる。

【0134】

$$D(1,1)=S(1)*Q(1,1,1)$$

$$D(2,1)=R(1)$$

次に、 $Q(j,i,k)$ を具体的に求める。ここで、 $n=2$ を定義に当てはめると、 j,i,k はいずれも1しか値をとらない。

【0135】

$c(j,i,k)$ は 1×1 行列である $U[1,1] \times (P[1,1])^{(j-1)}$ の i 行 k 列の値としたとき下記のように定義される。

【0136】

【数30】

$$c(j,i,k)=1 \text{ のとき } Q(j,i,k)=R(k)$$

$$c(j,i,k)=0 \text{ のとき } Q(j,i,k)=0$$

$$\begin{aligned} U[1,1] \times (P[1,1])^{(j-1)} &= U[1,1] \times (P[1,1])^0 \\ &= (1) \times E[1,1] \\ &= (1) \times (1) \\ &= (1) \end{aligned}$$

従って、 $c(1,1,1)$ は1であるから、 $Q(1,1,1)$ は $R(1)$ と定義される。

【0137】

以上から定義式は

$$D(1,1)=S(1)*R(1)$$

$$D(2,1)=R(1)$$

となる。変数 m を使用した形式では、

$$D(1, m+1) = S(m+1) * R(m+1)$$

$$D(2, m+1) = R(m+1)$$

となる。

【0138】

なお、分割数 $n=2$ の場合には、2個の分割データのうち、どちらか一方を取得しただけでは、元データ S を復元することはできず、2個のすべての分割データを取得して元データ S を復元することになる。

【0139】

なお、上記実施形態のデータ分割方法の処理手順をプログラムとして例えばCDやFDなどの記録媒体に記録して、この記録媒体をコンピュータシステムに組み込んだり、または記録媒体に記録されたプログラムを通信回線を介してコンピュータシステムにダウンロードしたり、または記録媒体からインストールし、該プログラムでコンピュータシステムを作動させることにより、データ分割方法を実施するデータ分割装置として機能させることができることは勿論であり、このような記録媒体を用いることにより、その流通性を高めることができるものである。

【0140】

上述してきたように、本実施例によれば、所定の定義式が元部分データと乱数部分データの排他的論理和からなるので、従来のように多項式や剰余演算を行う高速かつ高性能な演算処理能力を必要とせず、大容量のデータに対しても簡単な演算処理を繰り返して分割データを簡単かつ迅速に生成することができる。

【0141】

また、生成した複数の分割データのうち分割数よりも少ない数の分割データに対して定義式を適用することにより元データを復元するので、分割数よりも少ない任意の数 x の分割データで元データを復元でき、分割数から x を減算した数までの分割データを紛失したり破壊したとしても、元データを復元することができる。

【0142】

さらに、元データをネットワークを介して端末から受信し、この元データに対

して元部分データ、乱数部分データおよび分割部分データの生成処理を施して生成された複数の分割部分データをネットワークを介して保管サーバに送信して保管するので、多数のユーザが端末からネットワークを介してアクセスして分割処理を依頼することができ、共通化および経済化を図ることができる。

【0143】

【発明の効果】

以上説明したように、本発明によれば、元データを処理単位ビット長毎に区分けして複数の元部分データを生成し、複数の乱数部分データを生成し、各分割データを構成する各分割部分データを元部分データと乱数部分データの排他的論理和からなる所定の定義式に従って生成するので、従来のように多項式や剰余演算を用いることなく、コンピュータ処理に適したビット演算である排他的論理和演算を用いることにより高速かつ高性能な演算処理能力を必要とせず、大容量のデータに対しても簡単な演算処理を繰り返して分割データを簡単かつ迅速に生成することができるとともに、また分割データの保管に必要となる記憶容量も分割数に比例した倍数の容量よりも小さくすることができる。

【図面の簡単な説明】

【図1】

本発明の一実施形態に係るデータ分割方法を実施するデータ分割装置を含むシステム構成図である。

【図2】

図1に示す実施形態のデータ分割装置の分割数 $n=3$ の場合の分割処理を示すフローチャートである。

【図3】

16ビットの元データ S を8ビットの処理単位ビット長に基づいて分割数 $n=3$ で3分割する場合の各データと定義式および各分割部分データから元データを復元する場合の計算式などを示す表である。

【図4】

分割数 $n=3$ の場合の分割データ、分割部分データ、各分割部分データを生成する定義式を示す表である。

【図 5】

図 1 に示す実施形態のデータ分割装置の分割数 $n=4$ の場合の分割処理を示すフローチャートである。

【図 6】

分割数 $n=4$ の場合の分割データ、分割部分データ、各分割部分データを生成する定義式を示す表である。

【図 7】

分割数 $n=5$ の場合の分割データ、分割部分データ、各分割部分データを生成する定義式を示す表である。

【図 8】

図 1 に示す実施形態のデータ分割装置の分割数が n で処理単位ビット長が b である場合の一般的な分割処理を示すフローチャートである。

【図 9】

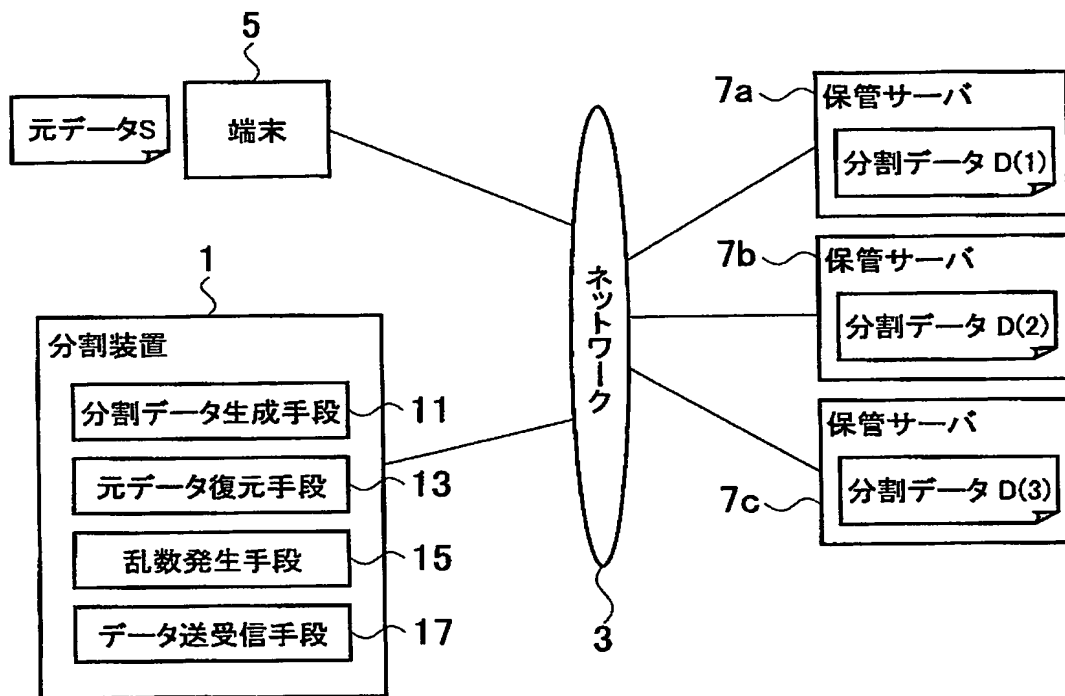
図 1 に示す実施形態のデータ分割装置の分割数が 2 である場合の分割処理を示すフローチャートである。

【符号の説明】

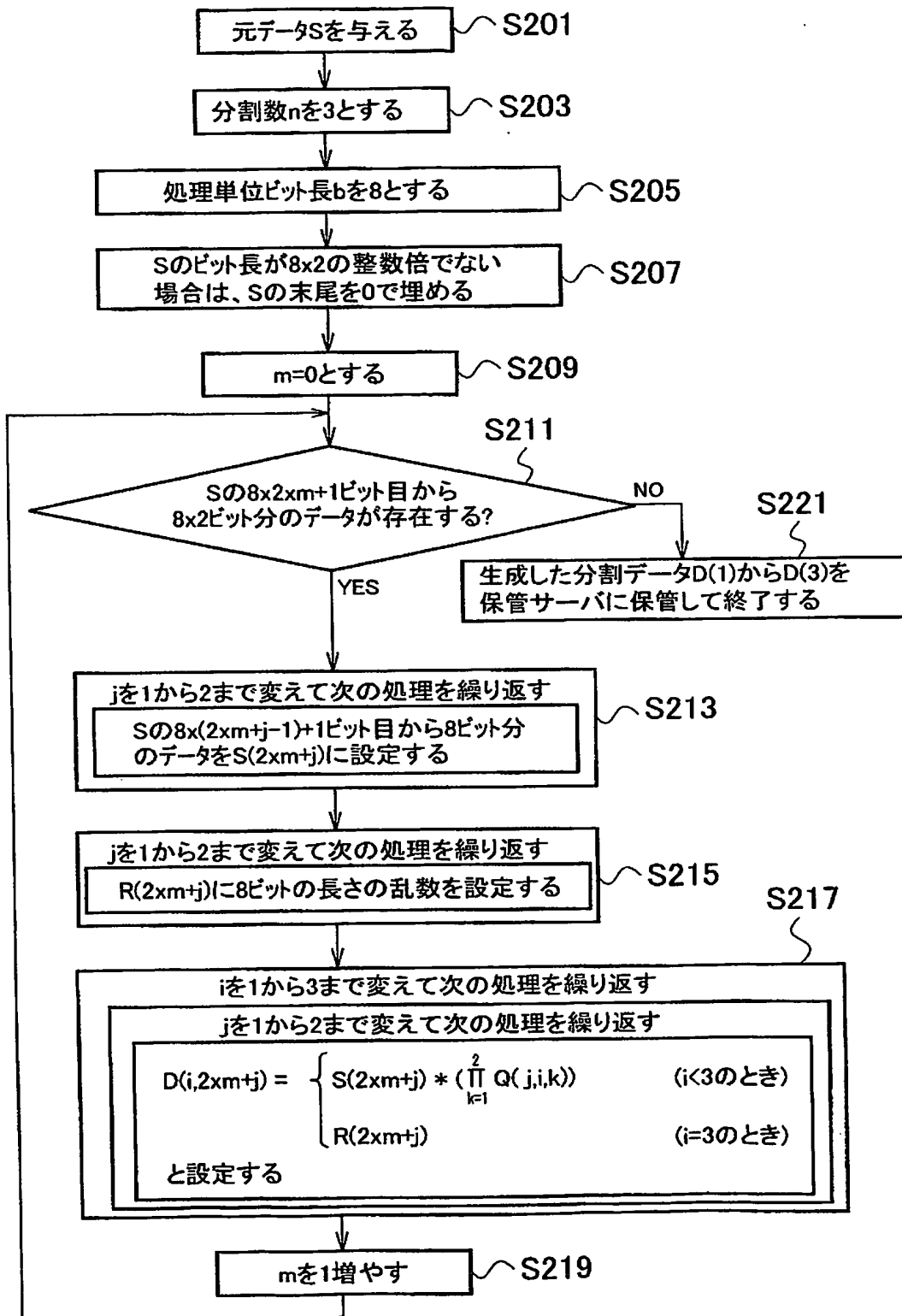
- 1 分割装置
- 3 ネットワーク
- 5 端末
- 7 a, 7 b, 7 c 保管サーバ
- 11 分割データ生成手段
- 13 元データ復元手段
- 15 乱数発生手段
- 17 データ送受信手段

【書類名】 図面

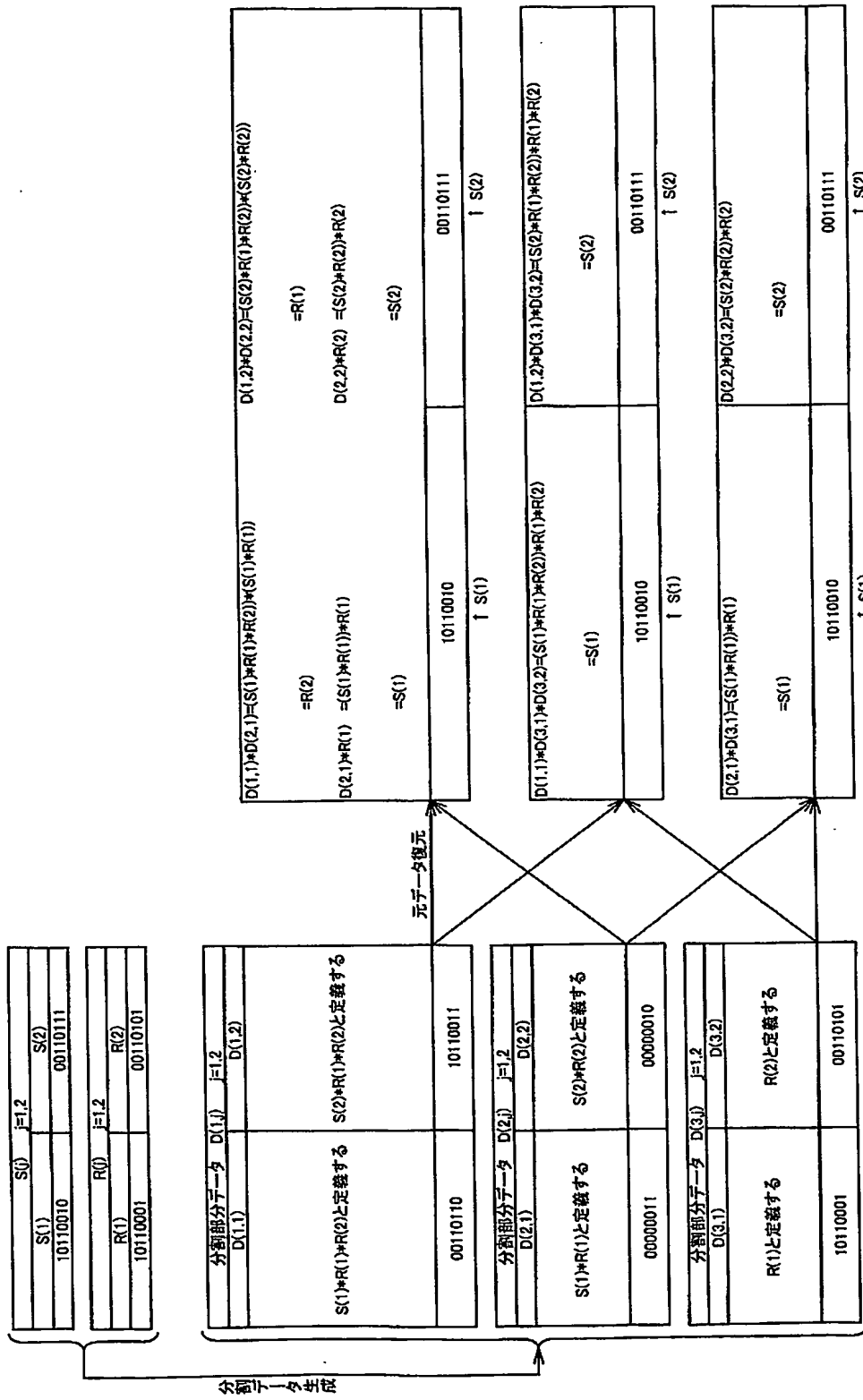
【図 1】



【図 2】



【図3】



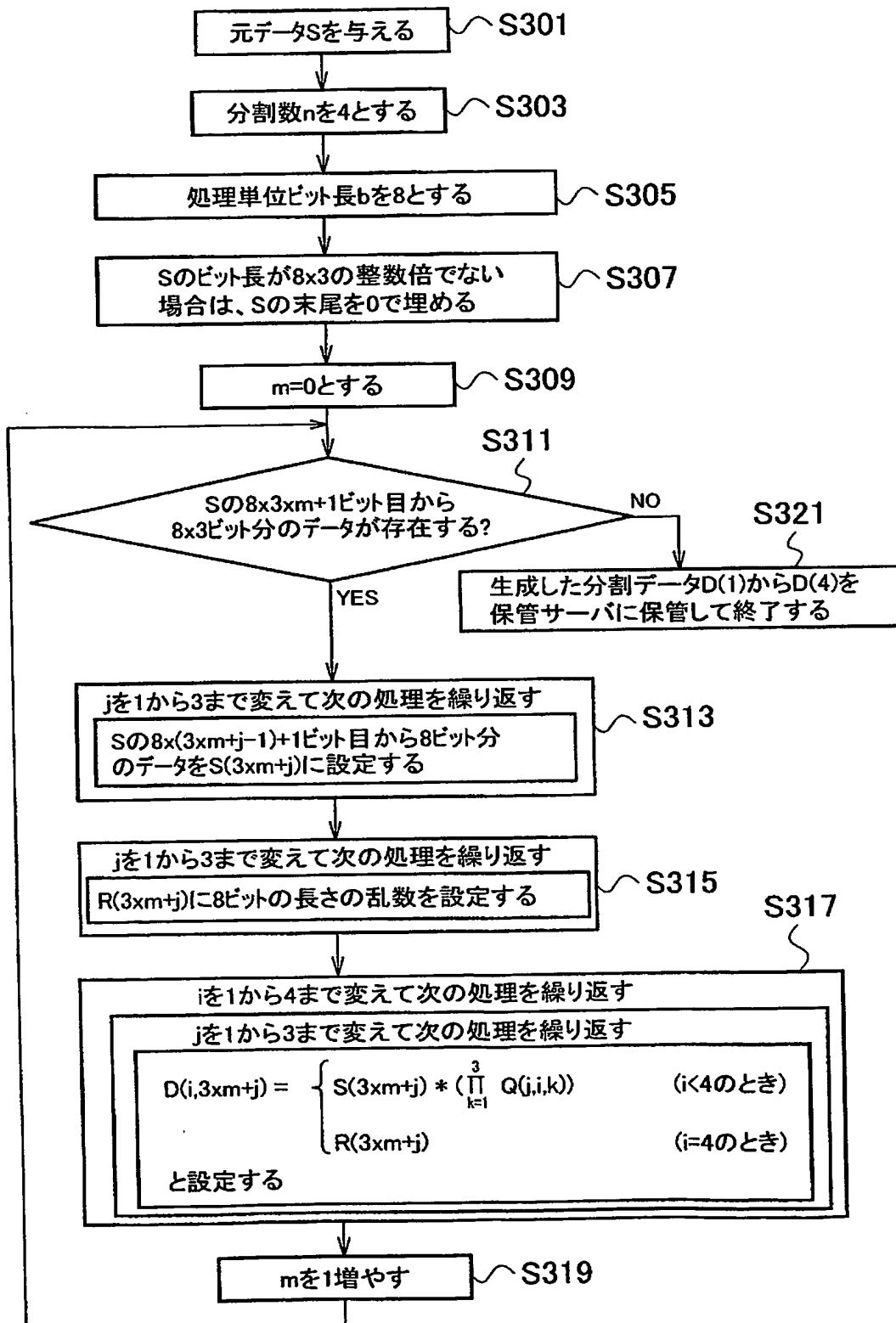
【図 4】

3分割 ($n=3$)

任意の2つの分割データから元データが復元可能。

		(mは $m>0$ の任意の整数)			→元データSの末尾まで続く	
jの値	1	2	...	$j=2 \times m+1$	$j+1$...
元データ S(i)	S(1)	S(2)	...	S(i)	S(j+1)	...
乱数 R(i)	R(1)	R(2)	...	R(i)	R(j+1)	...
分割部分データD(1,j)	$S(1)*R(1)*R(2)$	$S(2)*R(1)*R(2)$...	$S(i)*R(i)*R(j+1)$	$S(j+1)*R(i)*R(j+1)$...
分割部分データD(2,j)	$S(1)*R(1)$	$S(2)*R(2)$...	$S(i)*R(i)$	$S(j+1)*R(j+1)$...
分割部分データD(3,j)	R(1)	R(2)	...	R(i)	R(j+1)	...

【図 5】



【図6】

4分割 ($n=4$)

任意の3つの分割データ(取り方によっては2つの分割データ)から元データが復元可能。

jの値	1	2	3	...
元データ $S(j)$	$S(1)$	$S(2)$	$S(3)$...
乱数 $R(j)$	$R(1)$	$R(2)$	$R(3)$...
分割部分データD(1,j)	$S(1)*R(1)*R(2)*R(3)$	$S(2)*R(1)*R(2)*R(3)$	$S(3)*R(1)*R(2)*R(3)$...
分割部分データD(2,j)	$S(1)*R(1)*R(2)$	$S(2)*R(2)*R(3)$	$S(3)*R(1)*R(2)*R(3)$...
分割部分データD(3,j)	$S(1)*R(1)$	$S(2)*R(2)$	$S(3)*R(3)$...
分割部分データD(4,j)	$R(1)$	$R(2)$	$R(3)$...

1

→元データSの末尾まで続く

(m は $m>0$ の任意の整数)

$j=3 \times m+1$	$j+1$	$j+2$...
$S(j)$	$S(j+1)$	$S(j+2)$...
$R(j)$	$R(j+1)$	$R(j+2)$...
$S(j)*R(j)*R(j+1)*R(j+2)$	$S(j+1)*R(j)*R(j+1)*R(j+2)$	$S(j+2)*R(j)*R(j+1)*R(j+2)$...
$S(j)*R(j)*R(j+1)$	$S(j+1)*R(j+1)*R(j+2)$	$S(j+2)*R(j)$...
$S(j)*R(j)$	$S(j+1)*R(j+1)$	$S(j+2)*R(j+2)$...
$R(j)$	$R(j+1)$	$R(j+2)$...

1

【図 7】

5分割 ($n=5$)

任意の3つの分割データ(取り方によっては2つの分割データ)から元データが復元可能。

jの値	1	2	3	4	...
元データ $S(j)$	$S(1)$	$S(2)$	$S(3)$	$S(4)$...
乱数 $R(j)$	$R(1)$	$R(2)$	$R(3)$	$R(4)$...
分割部分データD(1,j)	$S(1)*R(1)*R(2)*R(3)*R(4)$	$S(2)*R(1)*R(2)*R(3)*R(4)$	$S(3)*R(1)*R(2)*R(3)*R(4)$	$S(4)*R(1)*R(2)*R(3)*R(4)$...
分割部分データD(2,j)	$S(1)*R(1)*R(2)*R(3)$	$S(2)*R(2)*R(3)*R(4)$	$S(3)*R(1)*R(3)*R(4)$	$S(4)*R(1)*R(2)*R(4)$...
分割部分データD(3,j)	$S(1)*R(1)*R(2)$	$S(2)*R(2)*R(3)$	$S(3)*R(2)*R(3)$	$S(4)*R(1)*R(4)$...
分割部分データD(4,j)	$S(1)*R(1)$	$S(2)*R(2)$	$S(3)*R(3)$	$S(4)*R(4)$...
分割部分データD(5,j)	$R(1)$	$R(2)$	$R(3)$	$R(4)$...

□

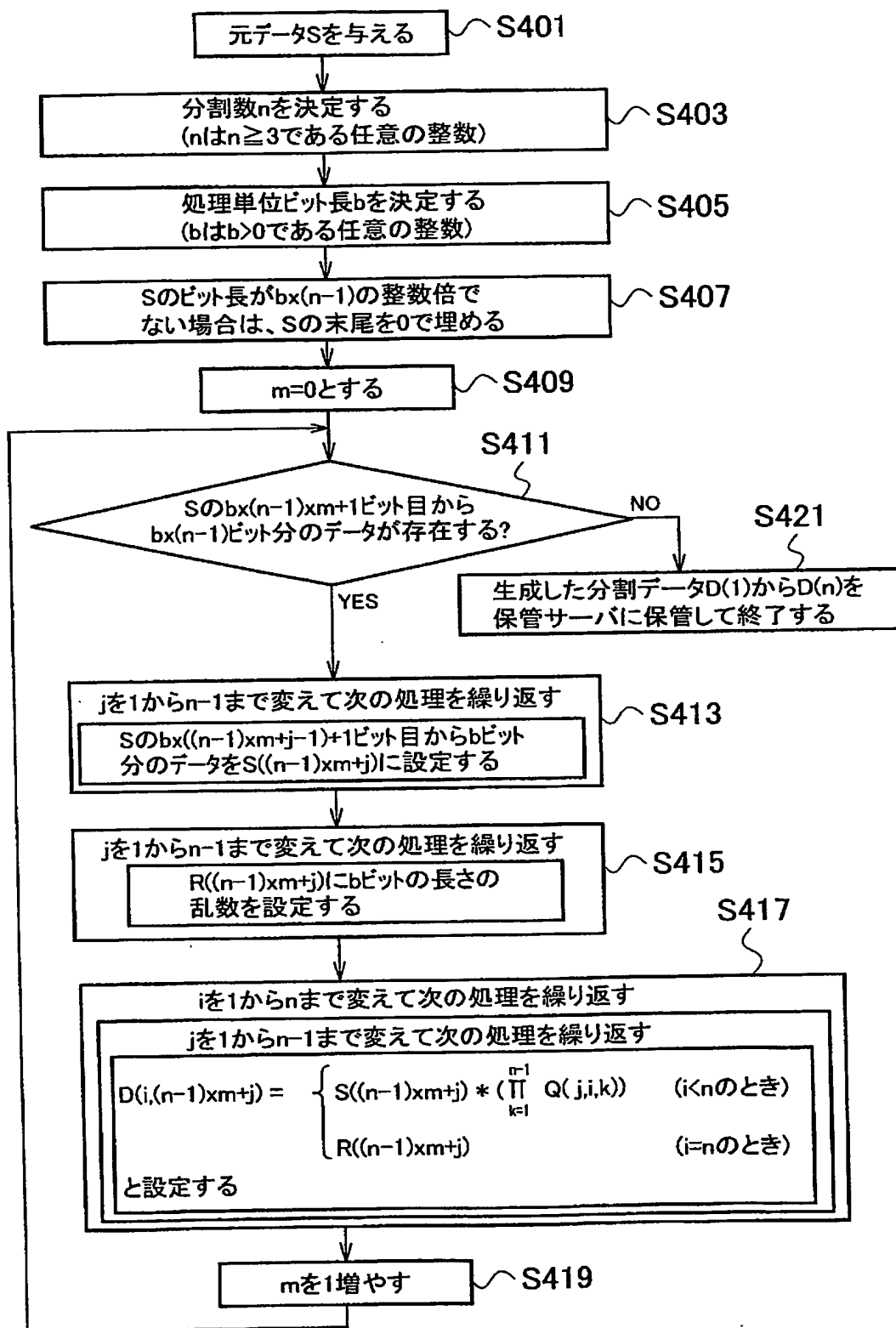
→元データSの末尾まで続く

(m は $m>0$ の任意の整数)

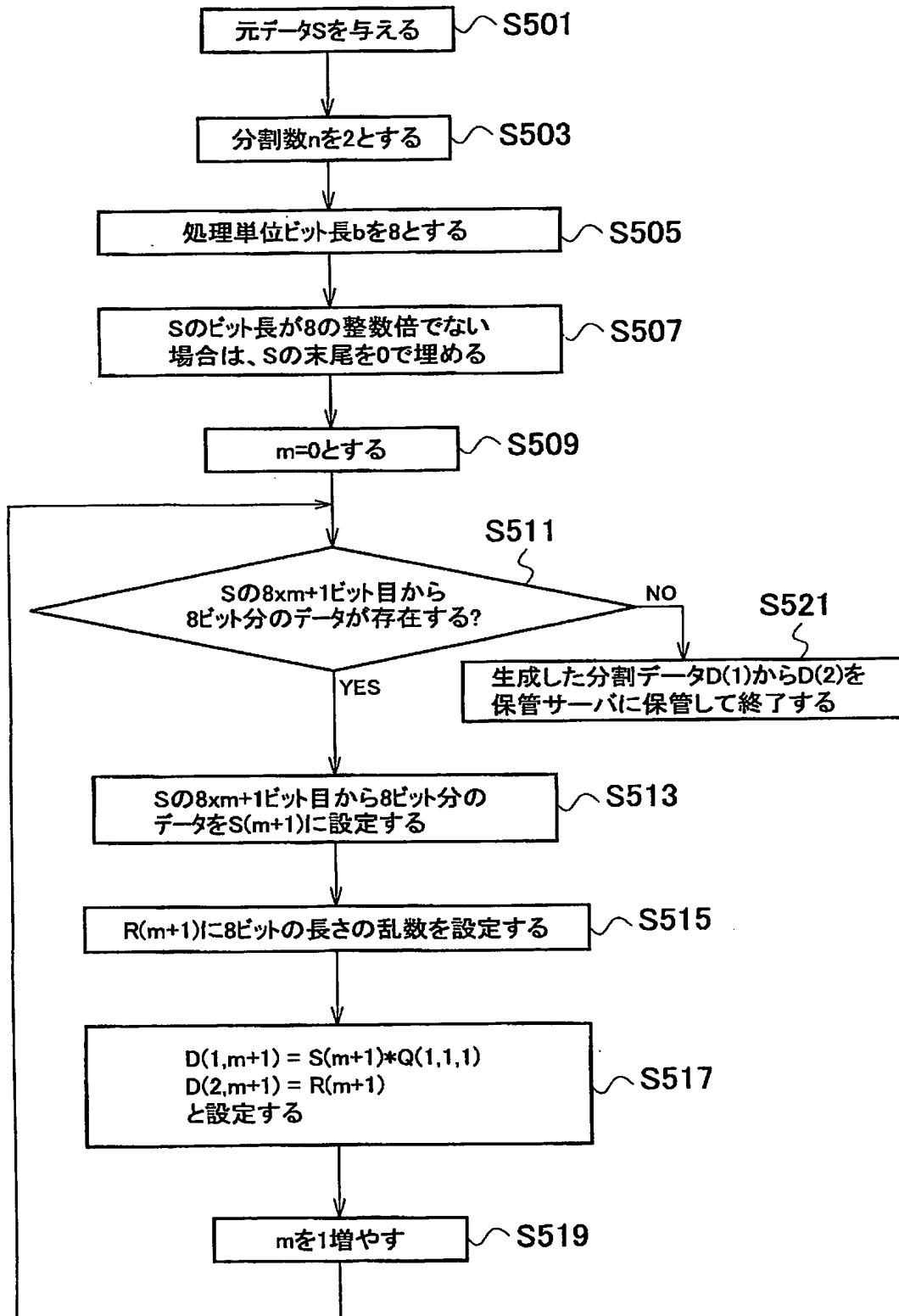
	$j-4 \times m+1$	$j+1$	$j+2$	$j+3$...
...	$S(j)$	$S(j+1)$	$S(j+2)$	$S(j+3)$...
...	$R(j)$	$R(j+1)$	$R(j+2)$	$R(j+3)$...
...	$S(j)*R(j)*R(j+1)*R(j+2)*R(j+3)$	$S(j+1)*R(j)*R(j+1)*R(j+2)*R(j+3)$	$S(j+2)*R(j)*R(j+1)*R(j+2)*R(j+3)$	$S(j+3)*R(j)*R(j+1)*R(j+2)*R(j+3)$...
...	$S(j)*R(j)*R(j+1)*R(j+2)$	$S(j+1)*R(j+1)*R(j+2)*R(j+3)$	$S(j+2)*R(j)*R(j)$	$S(j+3)*R(j)*R(j+1)*R(j+3)$...
...	$S(j)*R(j)*R(j+1)$	$S(j+1)*R(j+1)*R(j+2)$	$S(j+2)*R(j+2)$	$S(j+3)*R(j)$...
...	$S(j)*R(j)$	$S(j+1)*R(j+1)$	$S(j+2)*R(j+2)$	$S(j+3)*R(j+3)$...
...	$R(j)$	$R(j+1)$	$R(j+2)$	$R(j+3)$...

□

【図 8】



【図 9】



【書類名】 要約書

【要約】

【課題】 比較的簡単な処理により元データを効率的に分割し得るデータ分割方法および装置を提供する。

【解決手段】 元データS、分割数n、処理単位ビット長bを設定し（ステップS201-S205）、元データSを処理単位ビット長b毎に区分けして複数の元部分データS(j)を生成し（ステップS213）、複数の乱数部分データR(j)を生成し（ステップS215）、各分割データD(i)を構成する各分割部分データ(i, j)を元部分データと乱数部分データの排他的論理和からなる所定の定義式に従って生成する（ステップS217）。

【選択図】 図1

特願 2002-367608

出願人履歴情報

識別番号

[399035766]

1. 変更年月日

1999年 6月 9日

[変更理由]

新規登録

住 所

東京都千代田区内幸町一丁目1番6号

氏 名

エヌ・ティ・ティ・コミュニケーションズ株式会社